

Abstraction in EdTech vs. Computer Science Education Standards

Lawrence Tanimoto / CSTA Washington

[Lawrence Tanimoto | LinkedIn](#)

For the latest version of this presentation and related resources...

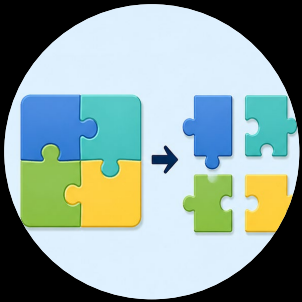
Visit the blog post on CSTAWA.org



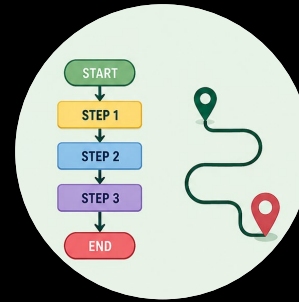
<https://tinyurl.com/abstraction-edtech-vs-cs>

Quick Audience Poll

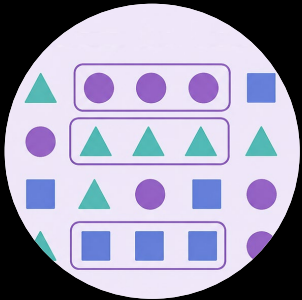
Of the four Computational Thinking concepts, which is the most difficult to teach?



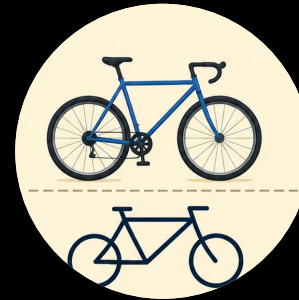
Decomposition



Algorithms



Pattern Recognition



Abstraction

Elementary Teacher Survey Results

Number of teachers (n=20) self-reporting struggle with...

PATTERN
RECOGNITION

0

DECOMPOSITION

1

ABSTRACTION

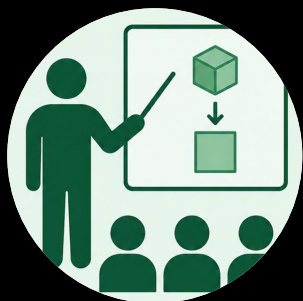
15

ALGORITHMS

1

Making Abstraction Concrete in the Elementary Classroom (SIGCSE 2024 Presentation (Hung, Vanderberg, Krause, Skuratowicz))

Quick Audience Poll



Do you feel that you teach abstraction in computer science well?



Do you feel that you understand abstraction in computer science well?

Abstraction is an Art

Not a dead artist

EDTECH VS. CS STANDARDS



Why this Matters?

- EdTech standards: technology literacy & fluency
- Computer Science standards: computational systems & problem solving
- Educators often blend the two together
- Students feel the difference later

CSTA K-12 CS Standards Progression Chart

Progression of Computer Science Teachers Association (CSTA) K-12 Computer Science Standards, Revised 2017					
Concept	Subconcept	Level 1A (Ages 5-7) <i>By the end of Grade 2, students will be able to...</i>	Level 1B (Ages 8-11) <i>By the end of Grade 5, students will be able to...</i>	Level 2 (Ages 11-14) <i>By the end of Grade 8, students will be able to...</i>	Level 3A (Ages 14-16) <i>By the end of Grade 10, students will be able to...</i>
Computing Systems	Devices	1A-CS-01 Select and operate appropriate software to perform a variety of tasks, and recognize that users have different needs and preferences for the technology they use. (P1.1)	1B-CS-01 Describe how internal and external parts of computing devices function to form a system. (P7.2)	2-CS-01 Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices. (P3.3)	3A-CS-01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects. (P4.1)
	Hardware & Software	1A-CS-02 Use appropriate terminology in identifying and describing the function of common physical components of computing systems (hardware). (P7.2)	1B-CS-02 Model how computer hardware and software work together as a system to accomplish tasks. (P4.4)	2-CS-02 Design projects that combine hardware and software components to collect and exchange data. (P5.1)	3A-CS-02 Compare levels of abstraction and interactions between application software, system software, and hardware layers. (P4.1)
	Troubleshooting	1A-CS-03 Describe basic hardware and software problems using accurate terminology. (P6.2, P7.2)	1B-CS-03 Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies. (P6.2)	2-CS-03 Systematically identify and fix problems with computing devices and their components. (P6.2)	3A-CS-03 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors. (P6.2)
Networks & The Internet	Network Communication & Organization		1B-NI-04 Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the Internet, and reassembled at the destination. (P4.4)	2-NI-04 Model the role of protocols in transmitting data across networks and the Internet. (P4.4)	3A-NI-04 Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing. (P4.1)
	Cybersecurity	1A-NI-04 Explain what passwords are and why we use them, and use strong passwords to protect devices and information from unauthorized access. (P7.3)	1B-NI-05 Discuss real-world cybersecurity problems and how personal information can be protected. (P3.1)	2-NI-05 Explain how physical and digital security measures protect electronic information. (P7.2)	3A-NI-05 Give examples to illustrate how sensitive data can be affected by malware and other attacks. (P7.2)
				2-NI-06 Apply multiple methods of encryption to model the secure transmission of information. (P4.4)	3A-NI-06 Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts. (P3.3)
Data & Analysis	Storage	1A-DA-05 Store, copy, search, retrieve, modify, and delete information using a computing device and define the information stored as data. (P4.2)	<i>Continuation of standard 1A-DA-05</i>	2-DA-07 Represent data using multiple encoding schemes. (P4.0)	3A-DA-09 Translate between different bit representations of real-world phenomena, such as characters, numbers, and images. (P4.1)
	Collection, Visualization, & Transformation	1A-DA-06 Collect and present the same data in various visual formats. (P7.1, P4.4)	1B-DA-06 Organize and present collected data visually to highlight relationships and support a claim. (P7.1)	2-DA-08 Collect data using computational tools and transform the data to make it more useful and reliable. (P6.3)	3A-DA-11 Create interactive data visualizations using software tools to help others better understand real-world phenomena. (P4.4)
	Inference & Models	1A-DA-07 Identify and describe patterns in data visualizations, such as charts or graphs, to make predictions. (P4.1)	1B-DA-07 Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea. (P7.1)	2-DA-09 Refine computational models based on the data they have generated. (P5.3, P4.4)	3A-DA-12 Create computational models that represent the relationships among different elements of data collected from a phenomenon or process. (P4.4)
Algorithms & Programming	Algorithms	1A-AP-08 Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks. (P4.4)	1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate. (P6.3, P3.3)	2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms. (P4.4, P4.1)	3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. (P5.2)
	Variables	1A-AP-09 Model the way programs store and manipulate data by using numbers or other symbols to represent information. (P4.4)	1B-AP-09 Create programs that use variables to store and modify data. (P5.2)	2-AP-11 Create clearly named variables that represent different data types and perform operations on their values. (P5.1, P5.2)	3A-AP-14 Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables. (P4.1)
	Control	1A-AP-10 Develop programs with sequences and simple loops, to express ideas or address a problem. (P5.2)	1B-AP-10 Create programs that include sequences, events, loops, and conditionals. (P5.2)	2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. (P5.1, P5.2)	3A-AP-15 Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made. (P5.2)
Practices		P1. Fostering an Inclusive Computing Culture P2. Collaborating Around Computing	P3. Recognizing and Defining Computational Problems P4. Developing and Using Abstractions	P5. Creating Computational Artifacts P6. Testing and Refining Computational Artifacts	P7. Communicating About Computing

The above link is invalid (06/11/2026)

CS Standards Covered By EdTech Standards

Progression of Computer Science Teachers Association (CSTA) K-12 Computer Science Standards, Revised 2017

Concept	Subconcept	Level 1A (Ages 5-7) <i>By the end of Grade 2, students will be able</i>	Level 1B (Ages 8-11) <i>By the end of Grade 5, students will be able</i>	Level 2 (Ages 11-14) <i>By the end of Grade 8, students will be able</i>	Level 3A (Ages 14-16) <i>By the end of Grade 10, students will be able</i>
Computing Systems	Devices	1A-CS-01 Select and operate appropriate software to perform a variety of tasks, and recognize that users have different needs and preferences for the technology they use. <i>(P1.1)</i>	1B-CS-01 Describe how internal and external parts of computing devices function to form a system. <i>(P1.2)</i>	2-CS-01 Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices. <i>(P3.3)</i>	3A-CS-01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects. <i>(P4.1)</i>
	Hardware & Software	1A-CS-02 Use appropriate terminology in identifying and describing the function of common physical components of computing systems (hardware). <i>(P1.2)</i>	1B-CS-02 Model how computer hardware and software work together as a system to accomplish tasks. <i>(P4.4)</i>	2-CS-02 Design projects that combine hardware and software components to collect and exchange data. <i>(P5.1)</i>	3A-CS-02 Compare levels of abstraction and interactions between application software, system software, and hardware layers. <i>(P4.1)</i>
	Troubleshooting	1A-CS-03 Describe basic hardware and software problems using accurate terminology. <i>(P6.2, P7.3)</i>	1B-CS-03 Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies. <i>(P6.2)</i>	2-CS-03 Systematically identify and fix problems with computing devices and their components. <i>(P6.2)</i>	3A-CS-03 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors. <i>(P6.2)</i>
Networks & The Internet	Network Communication & Organization		1B-NI-04 Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the Internet, and reassembled at the destination. <i>(P4.4)</i>	2-NI-04 Model the role of protocols in transmitting data across networks and the Internet. <i>(P4.4)</i>	3A-NI-04 Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing. <i>(P4.1)</i>
	Cybersecurity	1A-NI-04 Explain what passwords are and why we use them, and use strong passwords to protect devices and information from unauthorized access. <i>(P7.3)</i>	1B-NI-05 Discuss real-world cybersecurity problems and how personal information can be protected. <i>(P3.1)</i>	2-NI-05 Explain how physical and digital security measures protect electronic information. <i>(P7.2)</i>	3A-NI-05 Give examples to illustrate how sensitive data can be affected by malware and other attacks. <i>(P7.2)</i>
				2-NI-06 Apply multiple methods of encryption to model the secure transmission of information. <i>(P4.4)</i>	3A-NI-06 Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts. <i>(P3.3)</i>
					3A-NI-07 Compare various security measures, considering tradeoffs between the usability and security of a computing system. <i>(P6.3)</i>
					3A-NI-08 Explain tradeoffs when selecting and implementing cybersecurity recommendations. <i>(P7.2)</i>
Data & Analysis	Storage	1A-DA-05 Store, copy, search, retrieve, modify, and delete information using a computing device and define the information stored as data. <i>(P4.2)</i>	<i>Continuation of standard 1A-DA-05</i>		3A-DA-09 Translate between different bit representations of real-world phenomena, such as characters, numbers, and images. <i>(P4.1)</i>
	Collection, Visualization, & Transformation	1A-DA-06 Collect and present the same data in various visual formats. <i>(P1.1, P4.4)</i>	1B-DA-06 Organize and present collected data visually to highlight relationships and support a claim. <i>(P1.1)</i>	2-DA-08 Collect data using computational tools and transform the data to make it more useful and reliable. <i>(P6.3)</i>	3A-DA-11 Create interactive data visualizations using software tools to help others better understand real-world phenomena. <i>(P4.4)</i>
	Inference & Models	1A-DA-07 Identify and describe patterns in data visualizations, such as charts or graphs, to make predictions. <i>(P4.1)</i>	1B-DA-07 Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea. <i>(P7.1)</i>	2-DA-09 Refine computational models based on the data they have generated. <i>(P5.3, P4.4)</i>	3A-DA-12 Create computational models that represent the relationships among different elements of data collected from a phenomenon or process. <i>(P4.4)</i>
Algorithms & Programming	Algorithms	1A-AP-08 Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks. <i>(P4.4)</i>	1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate. <i>(P6.3, P3.3)</i>	2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms. <i>(P4.4, P4.1)</i>	3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. <i>(P5.2)</i>
	Variables	1A-AP-09 Model the way programs store and manipulate data by using numbers or other symbols to represent information. <i>(P4.4)</i>	1B-AP-09 Create programs that use variables to store and modify data. <i>(P5.2)</i>	2-AP-11 Create clearly named variables that represent different data types and perform operations on their values. <i>(P5.1, P5.2)</i>	3A-AP-14 Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables. <i>(P4.1)</i>
	Control	1A-AP-10 Develop programs with sequences and simple loops, to express ideas or address a problem. <i>(P5.2)</i>	1B-AP-10 Create programs that include sequences, events, loops, and conditionals. <i>(P5.2)</i>	2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. <i>(P5.1, P5.2)</i>	3A-AP-15 Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the tradeoffs. <i>(P5.2)</i>
					3A-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. <i>(P5.2)</i>

No Variables!

- **Variables are a CS concept not covered by EdTech**
- **Variables are foundational to abstraction**

Also missing: Storage, Culture

It's All About Abstraction

Actually, multiple layers of abstraction

Why Abstraction Matters



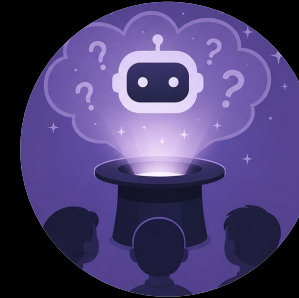
Every modern computing system runs on layers of abstraction



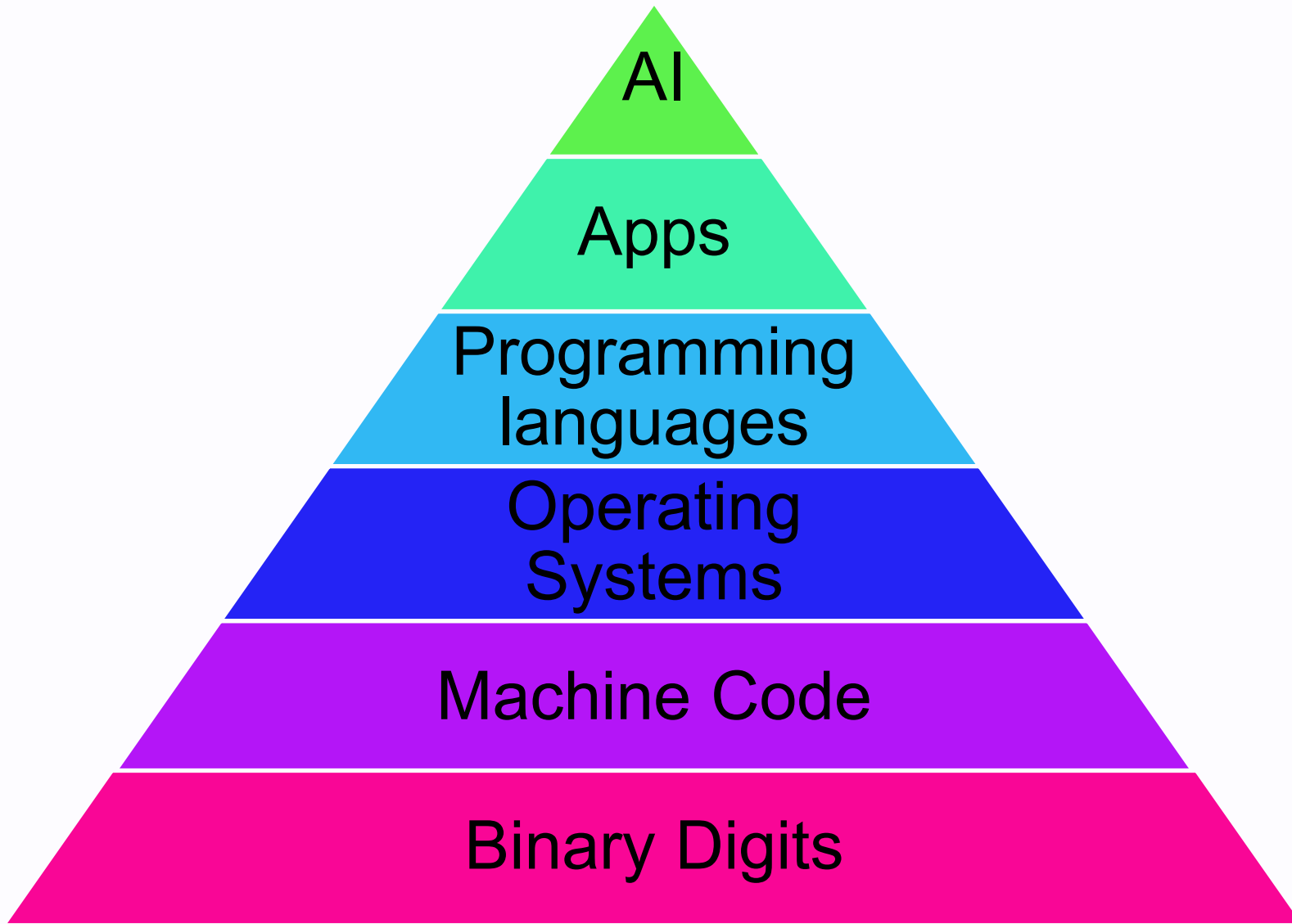
Without abstraction, modern software would be impossible



Each layer hides terrifying complexity below it



Without teaching abstraction, AI becomes magic



*"Any sufficiently advanced technology
is indistinguishable from magic."*

– Arthur C. Clark



Abstraction is the key to turning Magic into Knowledge.

The Bridge of Abstraction

Abstraction helps explain

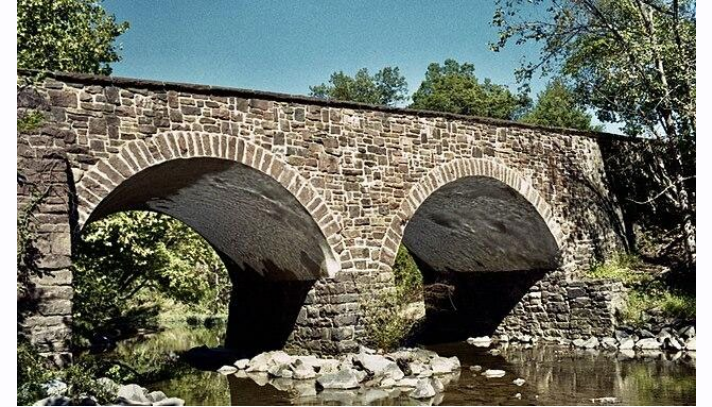
- why all students need to learn some computer science, but
- not everyone needs to major in it.

Abstraction in CS provides the literacy between

- math - precise but lacks context, and
- natural language - provides context but is imprecise

Abstraction provides the bridge language between

- humans - who often know what they want to do - even if they can't do it well, and
- computers - which don't know what they want to do but do it incredibly well if given the correct instructions.



EdTech Mostly Lives At The Top Layer of Abstraction

- Top layer: using applications
- Computer science asks HOW layers work.
- Each layer itself is not that complicated, but building upon multiple layers of abstraction makes magic happen
- Students deserve both

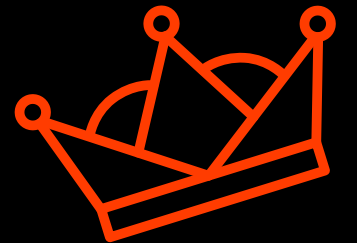
To better understand abstraction and how layers of abstraction make the computing world possible...

See the video on CSTAWA.org



<https://tinyurl.com/abstraction-edtech-vs-cs>

Thank You!



**This presentation
was created for use at**



2026 Annual Conference

